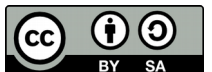


Workshop: GitHub 4 Newbies

25. 10. 2018 – DLE Universität Wien

Stefan Kasberger
@stefankasberger

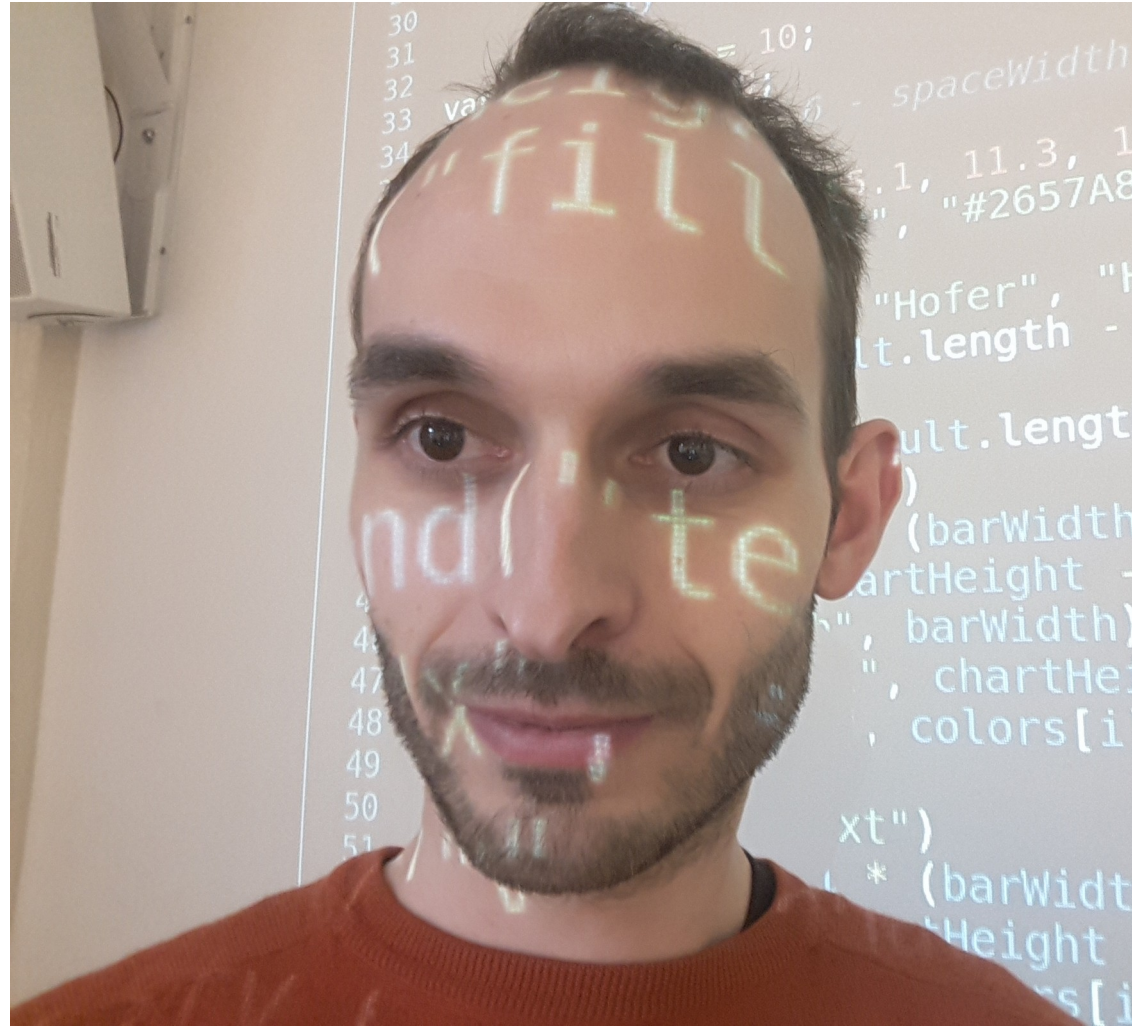


Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung-Weitergabe unter gleichen Bedingungen 4.0 International Lizenz.

EINLEITUNG

Stefan Kasberger

- Trainer & Data Scientist
- Open Science
- OANA
- Computational Social Science & Geography
- stefankasberger.at



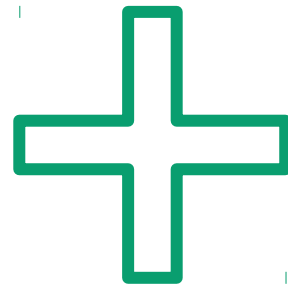
**„Wie man GitHub in der
Wissenschaft nutzen kann, um
1) die eigene Arbeit sichtbar
zu machen und 2) zum
Mitmachen einzuladen.“**

LEARNING GOALS

- Einführung in Versionsverwaltung mit Git.
- GitHub Account anlegen.
- Bei anderem GitHub Projekt mitwirken.
- Ein eigenes GitHub Projekt erstellen.



git



GitHub

BEISPIELE GITHUB

Lehrveranstaltung

Lehrveranstaltung Datengestützte Analysemethoden SS18

Lehrveranstaltungs-Repository für die Lehrveranstaltung *Datengestützte Analysemethoden* im Studiengang Journalismus und Public Relations (Sommersemester 2018).

- Lehrender: [Stefan Kasberger](#) (Email: edu@stefankasberger.at)
- LV Titel: Datengestützte Analysemethoden
- Studium: Journalismus und Public Relations
- Semester: 4., Sommersemester 2018
- Short URL: bit.ly/datenanalyse-ss18
- Lizenz: CC-BY 4.0 (sofern nicht explizit anders erwähnt)

VORBEREITUNG

- auf [GitHub](#) registrieren
- Text-Editor wie [Notepad++](#) oder [Atom Editor](#) installieren.
- [Libre Office](#) installieren.
- Markdown Editor installieren: [Remarkable](#) (nur Win und Linux).
- Markdown einarbeiten: [Mastering Markdown](#), [Writing on GitHub](#)

VORLESUNG

Inhalt	Folien	Daten
Einleitung LV	Einleitung	
Theoretische Grundlagen für das Arbeiten mit Daten	Daten	
Die vorbereitenden Datenverarbeitungs-Schritte in einer Datenanalyse.	Daten nutzen	Metadaten Umfrage Herrengasse , Metadaten Wetterstation
Analyse: Verteilungen, Metriken und Korrelation.	Daten auswerten	Beispiele Verteilungen , Beispiele Verteilungen für BoxplotR , Beispiele Korrelationen

iPython

codecov 66% pypl 47.0.1 build passed code helpers 100

IPython: Productive Interactive Computing

Overview

Welcome to IPython. Our full documentation is available on ipython.readthedocs.io and contains information on how to install, use and contribute to the project.

IPython versions and Python Support

IPython 7.0 requires Python version 3.4 and above.

IPython 6.x requires Python version 3.3 and above.

IPython 5.x LTS is the compatible release for Python 2.7. If you require Python 2 support, you **must** use IPython 5.x LTS. Please update your project configurations and requirements as necessary.

The Notebook, Qt console and a number of other pieces are now parts of *Jupyter*. See the [Jupyter installation docs](#) if you want to use these.

Development and Instant running

You can find the latest version of the development documentation on [readthedocs](https://readthedocs.org/projects/ipython/).

You can run IPython from this directory without even installing it system-wide by typing at the terminal:

```
$ python -m IPython
```

Or see the [development installation docs](#) for the latest revision on read the docs.

Documentation and installation instructions for older version of IPython can be found on the [IPython website](#)

IPython requires Python version 3 or above

Starting with version 6.0, IPython does not support Python 2.7, 3.0, 3.1, or 3.2.

For a version compatible with Python 2.7, please install the 5.x LTS Long Term Support version.

If you are encountering this error message you are likely trying to install or use IPython from source. You need to checkout the remote 5.x branch. If you are using git the following should work:

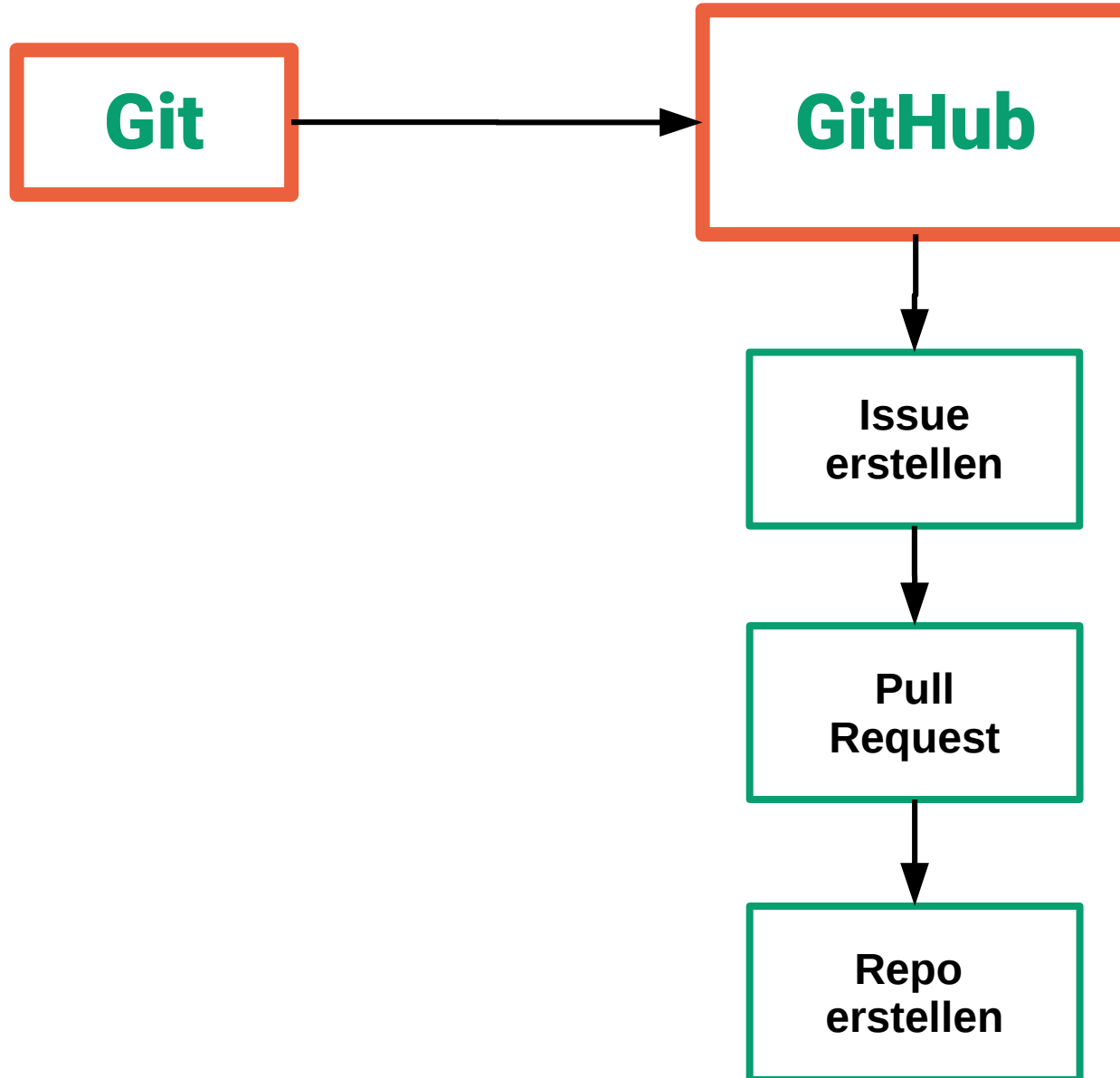
Data science

Thoughts and activities around [research data](#), with a focus on

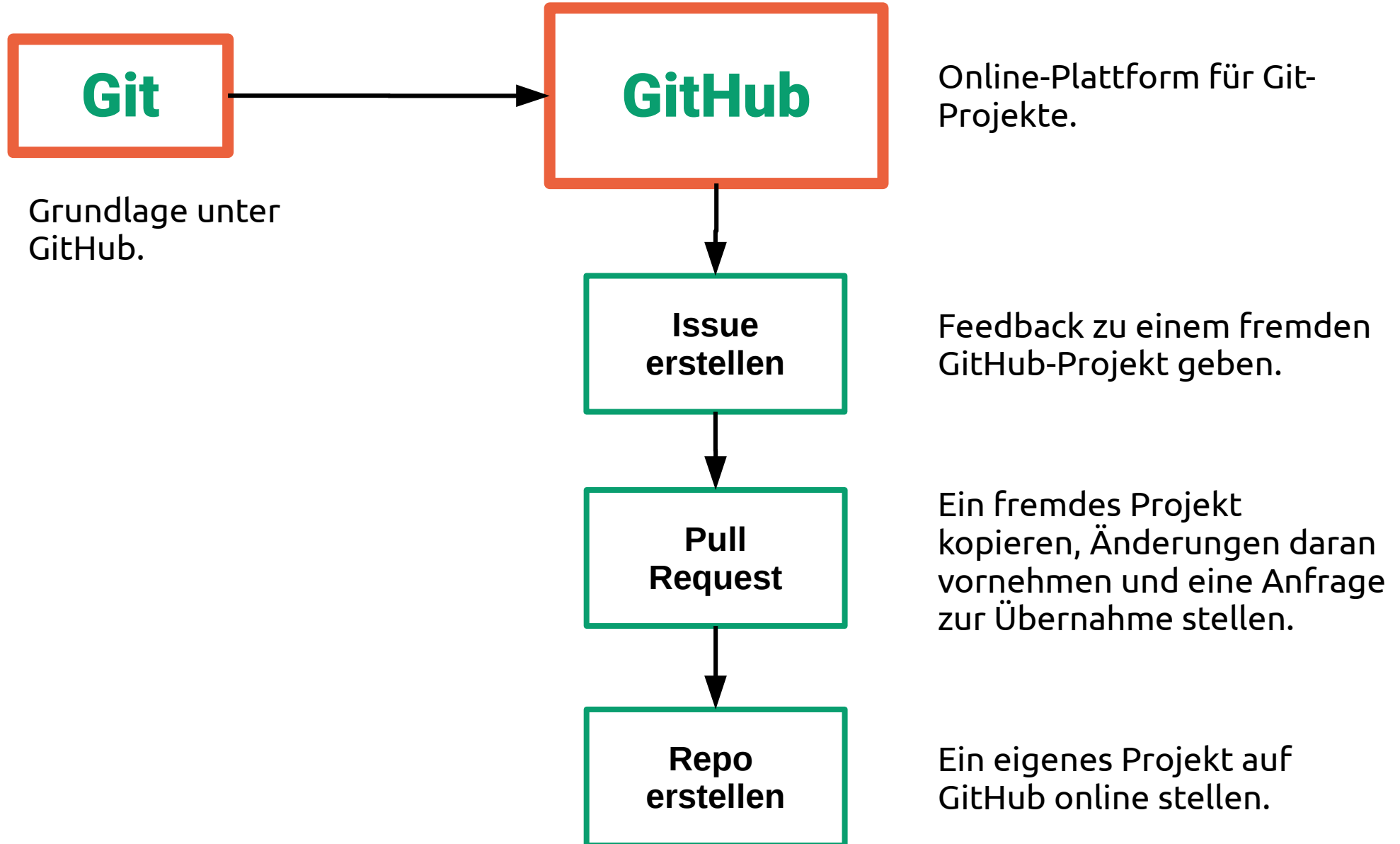
- Analysis of different kinds of open models for use in research funding.
 - [How can research funding be opened up to the benefit of research?](#)
 - [How can peer review be made more efficient?](#)
- Interfacing of the Commons with activities across institutional, disciplinary and international boundaries.
 - [How to conceptualize and implement a Commons infrastructure for research?](#)
 - [What policies govern open access to research outputs?](#)
 - [How does free and open-source software fit into the picture?](#)
 - [How can data best be described in order to lay the groundwork for interoperability across data sources?](#)
- Quantifying the reuse of various research objects, so as to gather information on what needs to be sustained.
 - [How are research objects being used and reused, and how can reuse be encouraged and quantified?](#)
 - [How best to cite data and software?](#)
 - [How to make data accessible for people with disabilities?](#)
 - [How to sustain infrastructure for data sharing?](#)
- Engaging communities with data, both experts and in the framework of citizen science projects.
 - [How best to integrate crowdsourcing components into data management?](#)
 - [How to make data management plans machine readable?](#)
 - [How can quality data journalism be supported?](#)

as well as a collection of further relevant resources in the [reading room](#).

ÜBERSICHT



ÜBERSICHT



bit.ly/2Pjlqd7

VORSTELLUNGSRUNDE

Pro Person ~ 10 Sekunden

- Name
- Disziplin
- Warum hier bzw. was erwartest du dir?

FRAGEN?

VERSIONSVER- WALTUNG MIT GIT

**Git ist eine
Versionsverwaltungs-
Software, welche Änderungen
an Dateien in einem Projekt
verwaltet.**

WARUM GIT?

- Verteiltes Arbeiten
- Kollaboratives Arbeiten
- Vollkommen nachvollziehbare und wiederherstellbare Geschichte der Dateien
- Open Source

NUTZUNG WISSENSCHAFT

- Software und Scripts
- Daten: CSV, XML, JSON
- LaTeX: Tex, BibTex,...
- Dokumentation: Projekt, Event, Protokolle,...
- Auch für Bilder, binäre Daten, Container solange nicht zu groß.

WIE FUNKTIONIERT GIT?

TEXT

- TXT
- Markdown
- HTML
- CSV/TSV
- JSON
- XML

NICHT-TEXT

- PDF
- JPG, PNG
- AVI, MPEG
- Microsoft Office

GIT WORKFLOW

COMMIT

- Commit Message: „init repo“

GIT WORKFLOW

Branch: master

COMMIT

- Commit Message: „init repo“

GIT WORKFLOW

Repository

Branch: master

COMMIT

- Commit Message: „init repo“

GIT WORKFLOW

Repository

Branch: master

COMMIT

- Commit Message: „init repo“



COMMIT

- „fix spelling error in README.md“

GIT WORKFLOW

Repository

Branch: master

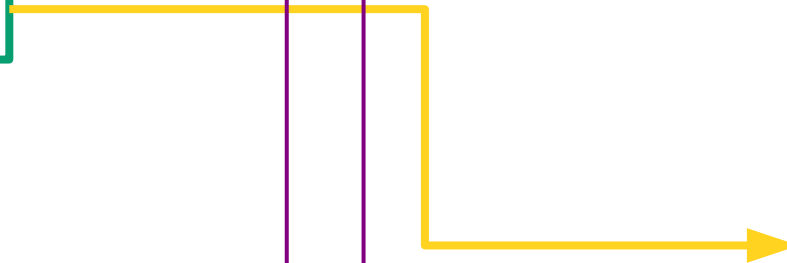
COMMIT

- Commit Message: „init repo“



COMMIT

- „fix spelling error in README.md“



Branch: workshop

GIT WORKFLOW

Repository

Branch: master

COMMIT

- Commit Message: „init repo“



COMMIT

- „fix spelling error in README.md“

Branch: workshop

Fork



GIT WORKFLOW

Repository

Branch: master

COMMIT

- Commit Message: „init repo“

**COMMIT**

- „fix spelling error in README.md“

**COMMIT**

- „add participation paragraph in README.md“

Branch: workshop

Fork

COMMIT

- „add WORKSHOP.md“

**COMMIT**

- „fix minor issues in WORKSHOP.md“

GIT WORKFLOW

Repository

Branch: master

COMMIT

- Commit Message: „init repo“



COMMIT

- „fix spelling error in README.md“



COMMIT

- „add participation paragraph in README.md“

Branch: workshop

Fork

COMMIT

- „add WORKSHOP.md“



COMMIT

- „fix minor issues in WORKSHOP.md“



GIT WORKFLOW

Repository

Branch: master

COMMIT

- Commit Message: „init repo“



COMMIT

- „fix spelling error in README.md“



COMMIT

- „add participation paragraph in README.md“

Branch: workshop

Fork

COMMIT

- „add WORKSHOP.md“



COMMIT

- „fix minor issues in WORKSHOP.md“

Pull Request



GIT WORKFLOW

Repository

Branch: master

COMMIT

- Commit Message: „init repo“



COMMIT

- „fix spelling error in README.md“



COMMIT

- „add participation paragraph in README.md“



COMMIT

- „merge event branch“

Branch: workshop

Fork

COMMIT

- „add WORKSHOP.md“

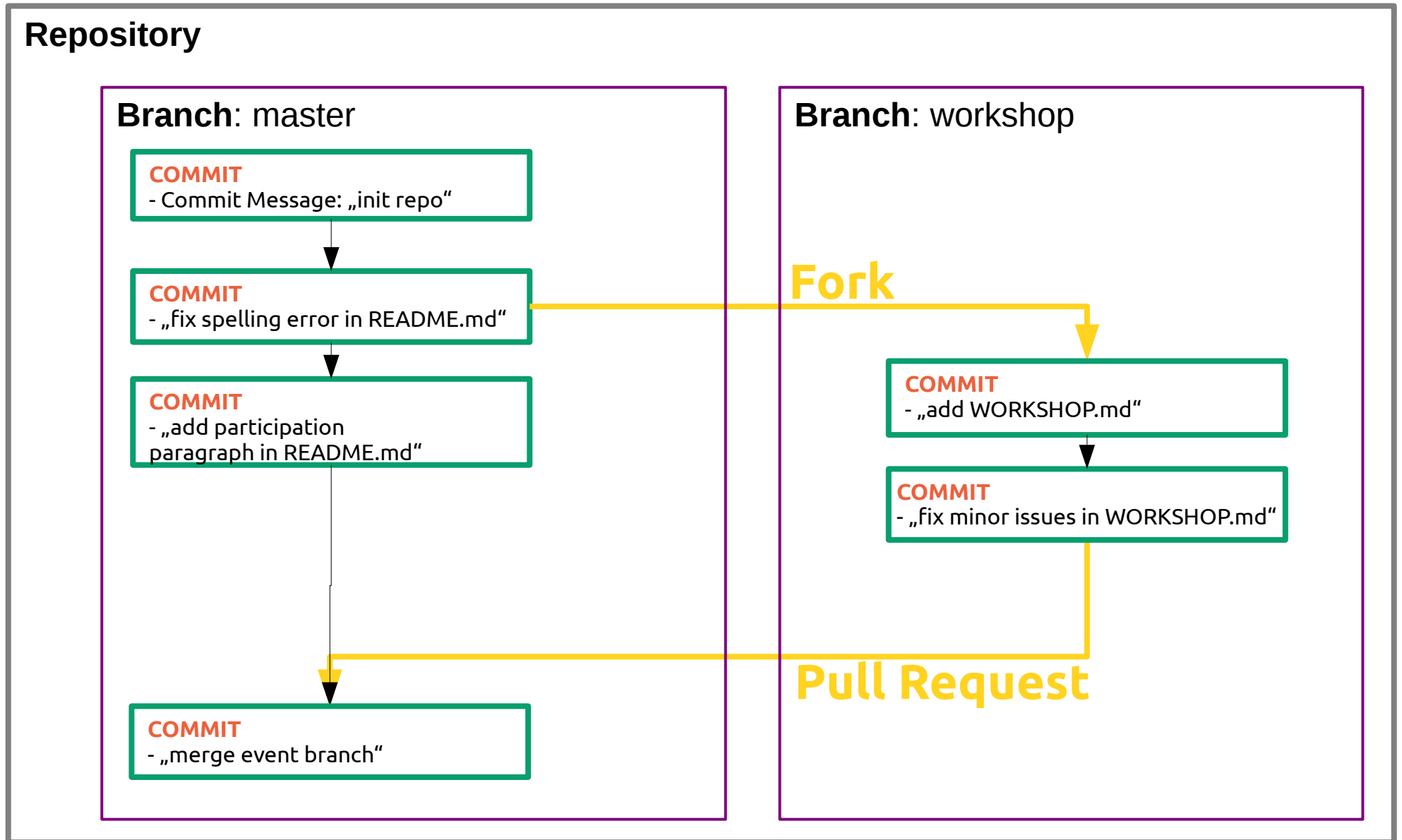


COMMIT

- „fix minor issues in WORKSHOP.md“



Pull Request



GIT WORKFLOW

Repository

Branch: master

COMMIT

- Commit Message: „init repo“

**COMMIT**

- „fix spelling error in README.md“

**COMMIT**

- „add participation paragraph in README.md“

**COMMIT**

- „merge event branch“

Merge

Branch: workshop

Fork**COMMIT**

- „add WORKSHOP.md“

**COMMIT**

- „fix minor issues in WORKSHOP.md“

Pull Request

BEISPIEL OFFICE-DOC VIA EMAIL

- Erster Draft (Commit)
- Zweiter Draft (Commit)
- Aussenden via Email (Fork)
- Antworten via Email (Pull Request)
- Zusammenfügen (Merge)

GITHUB

Online-Plattform, welche Git-Projekte hostet.

GITHUB

- Auf Git aufbauend mit zusätzlichen Features.
- Gratis für öffentliche Projekte
- Sehr viele Open Source Projekte
- GitHub ist das Facebook für Code

WARUM GITHUB?

- Daten speichern und online zur Verfügung zu stellen.
- Web-Oberfläche
- Funktionen: Issue Tracker, Reviewing, Wiki, GitHub Pages
- Vereinfachung durch Automatisierung
- Große Community
- Daten, Code und Dokumentation in einem Ordner.

NUTZUNG

- Dateigröße: begrenzt auf 100MB
- Projektgröße: Email bei ~ 1GB
- Aufpassen: Passwörter, API Tokens, etc.
- Englisch empfohlen
- Nicht als Datenspeicher per se gedacht (→ Zenodo)

SPEZIELLE RENDERINGS

- Bilder
- PDF
- Jupyter Notebooks
- CSV und TSV Dateien

ISSUE ERSTELLEN

**Einem anderen Projekt auf
GitHub Feedback geben.**

WOZU?

Verwalten von Tasks:

- Fehler (Bug)
- Verbesserungs-/Feature Wunsch
- Frage
- Hinweis

WOZU?

Verwalten von Tasks:

- Fehler (Bug)
- Feature Wunsch
- Hinweise

→ Repo-Owner bekommt eine Nachricht!

MARKDOWN

Auszeichnungs- Sprache

Einfach zu lesen und schreiben.

- Link: [TITEL](URL)
- Listenelement:
 - * ELEMENT
- Header1-6: # TITEL
- Doku
- Cheat-Sheet

Facebook Hidden Engagement Microservice

Simon Fraser University - Scholarly Communication Lab

Developed by [Stefan Kasberger](<http://stefankasberger.at>) and [Asura Enkhbayar](<https://github.com/Bubblbu>).

[![Build Status](<https://travis-ci.org/ScholCommLab/fhe-collector.svg?branch=master>)](<https://travis-ci.org/ScholCommLab/fhe-collector>)

[![Coverage Status](<https://coveralls.io/repos/github/ScholCommLab/fhe-collector/badge.svg?branch=master>)](<https://coveralls.io/github/ScholCommLab/fhe-collector?branch=master>)

Features

* Tests written in [pytest-flask](<http://pytest-flask.readthedocs.io/>) and executed with [Travis CI](<https://travis-ci.org/ScholCommLab/fhe-collector>). Test coverage by [pytest-cov](<https://pypi.org/project/pytest-cov/>) and [python-coveralls](<https://github.com/z4r/python-coveralls>), viewable on [coveralls.io](<https://coveralls.io/github/ScholCommLab/fhe-collector?branch=master>).
* auto-generated documentation through functions and class documentation with [sphinx](<http://www.sphinx-doc.org/>).

SETUP

This instructions are to setup the development environment, which is also the default environment.

Prerequisites

* registered Facebook app
* Configured server, which can deliver a Flask app

Download Flask app

Get the app on your computer, into your webserver's directory (e. g. vhost).

```
...  
cd /PATH/TO/VHOST  
git clone https://github.com/ScholCommLab/fhe-collector.git  
cd fhe-collector  
...
```

Setup virtualenv

Start the virtual environment to install the needed python packages.

```
...  
virtualenv --python=/usr/bin/python3 venv  
source venv/bin/activate  
pip install -r requirements.txt  
...
```

Install postgresSQL and create app database

If you want to use postgresSQL as your primary database, execute the following commands. If not, the app will default to SQLite. We recommend using postgresSQL as your standard database, cause we experienced some issues with SQLite during development.

GUTE ISSUES

- Keine Doppelgänger
- Titel soll den Beweggrund exakt und kurz beschreiben
- Freundliche und konstruktive Sprache verwenden
- Führe alle notwendigen Informationen an, um das Problem zu verstehen (z. B. Screenshots, Fehlermeldungen, Links).

ÜBUNG

- Neues Issue erstellen
- Text mittels zwei „Header 2“ in Blöcke unterteilen:
 - „Bemerkung“: Eigene Institution verlinken und Owner (@skasberger) direkt erwähnen.
 - „Wünsche“: Auflisten von 2-3 Wünschen, die man an das Projekt-Team stellt (je 1-2 Sätze) als Listenelemente.
- Kontrolle des Textes unter „Preview“.
- „Submit New Issue“ klicken!

→ Dokumentation nachsehen

Zeit: 3min

PULL REQUEST

**Änderungen an einem fremden
Projekt vornehmen, und
Anfrage zur Übernahme stellen.**

GIT WORKFLOW

Repository

Branch: master

COMMIT

- Commit Message: „init repo“



COMMIT

- „fix spelling error in README.md“



COMMIT

- „add participation paragraph in README.md“



COMMIT

- „merge event branch“

Merge

Branch: workshop

Fork

COMMIT

- „add WORKSHOP.md“



COMMIT

- „fix minor issues in WORKSHOP.md“



Pull Request



ÜBUNG

Pull Request zu Workshop Repo:

- Auf mein Repo gehen
- Forken
- Änderungen in teilnehmerinnen.md in Fork durchführen
- Repo Home gehen
- Pull Request machen
- Warten :)

REPO ERSTELLEN

**Ein eigenes Projekt auf GitHub
online stellen.**

INITIALISIERUNG

- Name: gut anhören und informativ
- Beschreibung: kurz und relevant (1 Satz)
- Repository Type: Public
- README → aktivieren
- Lizenz: MIT License

INHALT

- README.md
 - Titel, Beschreibung, Mitmachen, Kontakt, Urheberrecht, aktueller Projekt-Status, Inhalt & Struktur Repo
 - Default Homepage
- LICENSE
- Optional:
 - Code of Conduct (Contributor Covenant)
 - CONTRIBUTION.md
 - Wiki

ÜBUNG

- README.md anlegen
 - Titel
 - kurzer Beschreibungs-Text zum Projekt
 - Mitmachen Abschnitt (kurz mit SitznachbarIn besprechen).
 - Kontakt: Email, Website, Twitter, etc. (→ Links)
 - Unter welchem Urheberrecht steht der Inhalt
 - aktueller Projekt-Status
 - Inhalt & Struktur des Repo als Auflistung
- LICENSE: kontrollieren, ob da.

Zeit: 5min

OPEN, WAS NUN?

- Auf Issues und Pull Request zeitnah und positiv reagieren.
- Zugang verbessern: Sprache, Dokumentation, Code, aktuell halten
- Positive Kultur und Kommunikation → anderen das Gefühl geben, Sie können mit dem Projekt interagieren bzw. dazu beitragen.
- Bewerben bzw. einladen
- Verwendung des Repos forcieren
- DOI besorgen
- Lokal mit Git/GitHub arbeiten (Clients, Terminal)

EVALUIERUNG

**AUF
WIEDERSEHEN!**

Kontakt

@stefankasberger
stefankasberger.at

UrheberInnenrecht

Dieses Werk ist, sofern nicht explizit anders angegeben, lizenziert unter einer Creative Commons Namensnennung-Weitergabe unter gleichen Bedingungen 4.0 International Lizenz.

Urheber: Stefan Kasberger (2018).

Markenrecht

Alle in dieser Präsentation genannten Marken und Produktnamen sind eingetragene Marken-/Warenzeichen der jeweiligen Hersteller beziehungsweise Unternehmen.